BIRCH, STEWART, KOLASCH & BIRCH, LLP

INTELLECTUAL PROPERTY LAW
8110 GATEHOUSE ROAD
SUITE 500 EAST
FALLS CHURCH, VA 22042
U S A

(703) 205-8000

FAX: (703) 205-8050
(703) 698-8590 (G IV)

e-mail: mailroom@bskb.com
web: http://www.bskb.com

Date: May 11, 1999
Docket No.: 2950-0127P

Assistant Commissioner for Patents
Box PATENT APPLICATION
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): KANG, Seong-Cheol

For: METHOD FOR QUICKLY BOOTING A COMPUTER SYSTEM

Enclosed are:

X A specification consisting of 19 pages

X 6 sheet(s) of Formal drawings

X An assignment of the invention

X Certified copy of Priority Document(s)

X Executed Declaration X Original ___ Photocopy

___ A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27

___ Preliminary Amendment

___ Information Disclosure Statement, PTO-1449 and reference(s)

___ Other _____

The filing fee has been calculated as shown below:

			LARGE ENTITY			SMALL ENTITY	
FOR	NO. FILED	NO. EXTRA	RATE	FEE		RATE	FEE
BASIC FEE	********** ********** **********	********** ********** **********	***** ***** *****	$760.00	or	**** **** ****	$380.00
TOTAL CLAIMS	12 - 20 =	0	x18 =$	0.00	or	x 9 = $	0.00
INDEPENDENT	3 - 3 =	0	x78 =$	0.00	or	x 39 = $	0.00
MULTIPLE DEPENDENT CLAIM PRESENTED no			+260 = $	0.00	or	+130 = $	0.00
			TOTAL $	760.00		TOTAL $	0.00

_X_ A check in the amount of $ 800.00 to cover the filing fee and recording fee (if applicable) is enclosed.

___ Please charge Deposit Account No. 02-2448 in the amount of $_____. A triplicate copy of this transmittal form is enclosed.

___ No fee is enclosed.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37 C.F.R. 1.16 or under 37 C.F.R. 1.17; particularly, extension of time fees.

Respectfully submitted,

BIRCH, STEWART, KOLASCH & BIRCH, LLP

By _____

TERRY L. CLARK
Reg. No. 32,644
P. O. Box 747
Falls Church, Virginia 22040-0747

(703) 205-8000
TLC/aam

METHOD FOR QUICKLY BOOTING A COMPUTER SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a method of and apparatus for booting a personal computer system and, more particularly, for quickly booting a computer system, in which a boot configuration information is created and saved in a disk for future boot, and the saved boot configuration information is reused upon the request of the subsequent boot.

2. Description of the Related Art

Figure 1 shows a block diagram of the system architecture for a conventional personal computer system, comprising a central processing unit (CPU) 1; a read only memory (ROM) 2 for

permanent storage of basic input output system (BIOS) and the initial states of internal devices, a random access memory (RAM) 3 for temporary storage of information; a micro computer (MICOM) 4 for controlling peripheral devices such as a keyboard input device, a mouse input device, and a power supply; a hard disk (HDD) 8 for providing a secondary information storage; a disk controller 5 for controlling HDD; a video output display 6 for displaying information; and a power supply 7. When power is applied to the computer system, the computer system starts to be booted to load an operating system (OS) and thus is brought into a known useful state in which application programs can be executed. This procedure is generally called "booting". An operating system is a software that provides resource management on a computer system, including basic tasks such as process execution, memory management, and file management. Examples are MS-DOS, Windows95, OS/2, and UNIX. Execution of user applications is based on these basic functions of the operating system.

The boot process of an IBM PC in which MS-DOS operating system is already installed is as follows. When a user turns the personal computer power switch on or presses a reset button, a power-on self test (POST) is performed by ROM BIOS codes to diagnose each component of the personal computer. Next, a file called MSDOS.SYS is loaded and executed, and another file called IO.SYS is then loaded and executed to perform certain

preliminary functions related to management of such peripheral devices as keyboard, disk, and display. And then, a command preprocessor or COMMAND.COM is loaded into a memory that receives, interprets and executes user commands. A file called CONFIG.SYS that specifies devices possibly connected to the personal computer is loaded and ASCII statements contained therein are executed to load device drivers and initialize them. Finally, another ASCII file called AUTOEXEC.BAT is loaded and then programs that is listed therein are executed, thereby preparing the personal computer for use.

There two kinds of boots; "cold boots" and "warm boots", which rely on the state of the computer system when the boot operation is requested. A "cold boot" is performed when power is applied to computer or a reset button is pressed. When an operating system is loaded in memory already and the computer system is powered on already, a user may request a "warm boot" by entering a predefined sequence of key strokes, e.g., <Ctrl>+<Alt>+<Del>. The BIOS codes consist of a plurality of computer routines for controlling devices such as system clock, video output display, disk driver, and keyboard and thus provide a low-level interface to these devices. The BIOS is generally stored in a Flash ROM.

Shortly after power on or a reset button is pressed, the CPU begins executing the ROM BIOS codes. The BIOS codes for POST are, first, executed to diagnose and initialize devices

attached to the computer system and obtain the status of the devices.

When a "warm boot" is requested or a reset button is pressed, it is desirable that the time required for the boot process is reduced to force the computer into a ready state as quickly as possible. The boot process is usually called "quick boot", which is achieved by simplifying some device diagnosis processes or loading the device status information that was obtained at the preceding boot time from a storage medium such as disk. Because the quick boot means a boot process in which some POST operations, e.g., memory test are skipped, the quick boot is generally referred to as "quick post".

FIG. 2 is a flowchart of the quick POST in an IBM personal computer system in which Windows95 is installed according to the conventional art. When the computer system is powered on or a reset button is pressed (S11), the Windows95 is loaded into a memory after execution of a normal POST process (S12). To be specific, once the POST process is performed, ASCII statements in CONFIG.SYS and AUTOEXEC.BAT are executed sequentially and WIN.COM is then executed to load Windows95. While Windows 3.1, a previous version of Windows95, is loaded after the personal computer is booted on the basis of MS-DOS, Windows95 installed PC is booted and Windows95 user interface is provided directly.

Once the boot operation is completed, a basic boot information is saved to a disk for future quick POST process

(S13). After that, if a user requests a "quick boot" to reboot the personal computer (S14), the above-mentioned quick POST process is performed to reduce the time needed to complete a normal POST process. As another method, the POST process execution is skipped by using a basic boot information that was created and saved in a disk immediately after the preceding POST process is completed.

However, the conventional quick boot relies on the POST process, e.g., the omission of memory test. In other words, in the conventional quick booting method, the same operations as those of normal boot process are still performed after the quick POST process. Therefore, in case where there are a lot of ASCII statements in CONFIG.SYS and AUTOEXEC.BAT, the quick boot of the conventional art is not effective to reduction of the boot time.

According to the conventional booting method, in Windows95 installed personal computer system, working environment or all information stored in memory are saved to a disk for the subsequent quick boot. If memory size is larger than 32 MB, the amount of data to be saved to the disk becomes too large. As a result, the subsequent booting by reloading the saved data into the memory may be even slower than a normal boot.

SUMMARY OF THE INVENTION

It is therefore a primary object of the present invention

to provide a method and apparatus that significantly reduces the time required for boot process after a POST operation by using a boot configuration information on memory and the attached devices that was created and saved in a disk in the preceding boot process, and thereby skipping execution of statements in an initial device configuration file and an automatic batch file.

To achieve the object, the present invention provides a method for quickly booting a personal computer system, comprising the steps of performing a POST operation when the system is powered on or a reset button is pressed; checking if a boot configuration information that was created in the preceding boot process exists in a disk; saving the boot configuration information to the disk after execution of a POST operation on the basis of the checking result; and loading a graphic user interface (GUI) program.

The method for quickly booting a computer system according to the present invention is also characterized in that it comprises the steps of performing a POST operation when the system is powered on or a reset button is pressed; restoring a boot configuration information by using the boot configuration information that has been saved in a disk; and loading a GUI program.

According to the quick booting method of the present invention, after ASCII statements listed in an initial device

configuration file and an automatic batch file are executed, a boot configuration information that is resident in a memory, i.e., the status of devices and the contents of memory are saved into a disk. After that, when a reboot is requested, a computer system can be booted quickly by using the stored boot configuration information, without execution of the initial device configuration file and the automatic batch run file.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention, illustrate the preferred embodiment of this invention, and together with the description, serve to explain the principles of the present invention.

In the drawings:

FIG. 1 is the system architecture of a general personal computer system;

FIG. 2 is a flowchart showing the conventional method for quick POST operation in a Windows95-installed personal computer;

FIG. 3 is a flowchart showing a method for a quick boot according to an embodiment of the present invention;

FIG. 4 is a flowchart showing a method for saving a boot configuration information after execution of POST operation in a Windows95-installed personal computer according to an

embodiment of the present invention;

FIG. 5 is a flowchart showing a method for restoring a stored boot configuration information in a Windows95-installed personal computer according to an embodiment of the present invention;

FIG. 6 is a flowchart showing a method for saving the contents of memory into a disk according to an embodiment of the present invention; and

FIG. 7 is a flowchart showing a method for restoring the contents of memory according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFFERRED EMBODIMENT

The preferred embodiments of the present invention will be described below in detail referring to the accompanying drawings.

FIG. 3 is a flowchart of the quick boot process in an IBM personal computer system according to an embodiment of the present invention. The flow proceeds to step S21, in which a POST operation is performed when a computer system is powered on or a reset button is pressed. A normal boot process of an operating system, e.g., Windows95 is then executed (S22). Next, a boot configuration information, i.e., the contents of memory and the status of the attached devices that was created and has been resident in a memory since execution of the POST operation

is saved to a disk (S23). A RAM-resident program is called by a software interrupt (INT in general) in modified ROM BIOS codes so as to save the boot configuration information into a disk for future boots. Since then, when a reboot is requested (S24), the POST operation is performed and then the saved boot configuration information is retrieved from the disk in order to complete the reboot process (S25). If it is determined that either CONFIG.SYS or AUTOEXEC.BAT was changed (S26), the changed two files are loaded into a memory and are then executed to form a new boot configuration information, which will be saved to the disk again for the subsequent boot. In this way, if CONFIG.SYS and AUTOEXEC.BAT are not changed, they do not need to be loaded and executed when a computer system is booted, resulting in a quick boot.

FIGS. 4 and 5 are flowcharts respectively showing methods for saving and restoring a boot configuration information in a Windows95-installed IBM personal computer according to an embodiment of the present invention.

The method for saving the boot configuration information to a disk will be described now in detail referring to FIG. 4. When power is turned on or a reset button is pressed (S31), a cold boot or warm boot is requested. The POST operation is, first, executed (S32) and then an INT 19h service routine is called to load an operating system (S33). By calling the INT 19h, control is passed to a bootstrap loader which loads the

operating system into a memory to prepare the personal computer for use.

Next, CONFIG.SYS is loaded into the memory and statements therein are executed. CONFIG.SYS includes ASCII statements describing the size of disk buffer, the number of files that can be opened simultaneously, the names of device drivers needed to control devices attached to the computer system, and so on. After executing CONFIG.SYS, another ASCII file called AUTOEXEC.BAT is loaded into the memory. The file names of programs that a user wants to run automatically at the boot time are listed therein, and the programs are executed (S34).

Next, a RAM-resident program is activated to replace an original INT 2Fh service routine in the ROM BIOS codes. To do this, the interrupt vector for INT 2Fh is substituted for the address of the RAM-resident program (S35). Next, WIN.COM is executed to load Windows95 into the memory (S36). The INT 2Fh service routine is called by using software system management interrupt (software SMI) during the execution of WIN.COM. At the interrupt point, the contents of a particular register is sent to the RAM-resident program and then performs a prescribed function associated with the register contents (S37).

If the register contents is a predetermined value, e.g., 1605H, the RAM-resident program checks if there is a file that contains the boot configuration information in a disk (S38) and saves the current boot configuration information to the disk,

if not (S39). WIN.COM is then executed to load a GUI program of Windows95 into the memory (S41), providing a user with Windows95 interface (S42). It should be noted that the boot configuration information is saved to the disk immediately before Windows95 loads device drivers into a memory, i.e., an extended memory is used to load GUI program of Windows95.

The operation of saving the boot configuration information to a disk (the step S39 of FIG. 4) is described in detail with reference to a flowchart of FIG. 6. The contents of memory block of a predetermined size are, first, examined and are then saved to the disk if the memory block is satisfied with a predetermined criterion. An address of the memory block is saved to the disk, as well. To be specific, if it is determined that the boot configuration information resident in a memory needs to be saved to the disk (S71), the INT 2Fh service routine checks if a memory segment of 64 KB is filled with '0', while scanning every memory segment (S72). If not, the contents of the memory segment are saved to the disk (S73), together with its address (S74). The memory segment is treated as a memory accessing unit, which is 64 KB in size in the IBM personal computer system. And the boot configuration information to be saved is approximately 7MB in size, which is composed of 1MB for saving the software SMI, 4MB for the video memory, and 2MB for saving a memory area in which the interrupt vector table and some crucial programs for system management are resided.

The next time the computer system is powered on or reset, the saved boot configuration information is used to boot the computer system. The method for restoring the boot configuration information will be described now in detail referring to FIG. 5.

Once power is turned on or reset button is pressed (S51), a quick POST operation including skip of memory test is executed (S52), and then it is checked whether or not there is any boot configuration information that has been saved to a disk in the preceding boot process (S52-1). If it is determined that a boot configuration information exists, the operation for its restoration is performed (S53).

The process for restoring the boot configuration information is described in detail referring to a flowchart of FIG. 7. First, it is checked whether or not a current boot configuration has been changed based on the restored boot configuration information. If there is any change in the boot configuration, commands that are usually executed at the boot time, for example, commands for initial setup device configuration are executed and then a newly formed boot configuration information is saved to the disk for future boot. Specifically, when a computer system is resumed, it is checked if the boot configuration information will be restored (S81). If it is determined that the boot configuration information is restored, the contents of memory segments, addresses of which

was saved before in the disk, are copied to the memory at their own addresses (S82). Other memory segments than the restored memory segments become filled with '0' (S83). The reason why the contents of those segments are not restored is that they are set to all '0's during the BIOS POST operation.

Once restoration of the contents of those memory segments is completed, it is checked if CONFIG.SYS and AUTOEXEC.BAT was changed (S54). If it is determined that either CONFIG.SYS and AUTOEXEC.BAT was changed, the bootstrap loader, the INT 19h service routine is called (S55), and then the both ASCII files are loaded into a memory to execute statements therein (S56). Next, the RAM-resident program is activated to replace the original INT 2Fh service routine (S57).

Next, WIN.COM is executed to load Windows95 into the memory (S58). The INT 2Fh service routine is called through the software SMI during the execution of WIN.COM (S59) and thus the RAM-resident program is executed. The RAM-resident program, first, checks the contents of a particular register and then, if it is matched with a predetermined value, saves the current boot configuration information that is resident in the memory to the disk (S61). The method for saving the contents of memory where the boot configuration information is resided is the same as that shown in FIG. 6. Next, control is passed to WIN.COM again and thus Windows95 GUI is set up (S62), thereby preparing the computer system for use (S63).

The reason why the boot configuration information should be restored before Windows95 loads Windows95-dedicated device drivers is to reduce the amount of data on the disk that must be copied to a memory at the boot time. Because the extended memory has been not used yet at that time, even if the size of total memory is 512MB, memory of about 7MB only is saved to and restored from the disk, according to the methods shown in FIGS. 6 and 7. Therefore, the execution of CONFIG.SYS and AUTOEXEC.BAT can be skipped by restoring the contents of memory and the status of devices based on the boot configuration information, thereby reducing the boot time significantly.

Though the description hereinbefore may refer to terms commonly used in describing particular computer systems and software, such as IBM personal computer and Windows95 operation system, the concepts equally apply to other systems and software.

The foregoing is provided only for the purpose of illustration and explanation of the preferred embodiments of the present invention, so changes, variations and modifications may be made without departing from the spirit and scope of the invention.

What is claimed is:

1. A method for booting a computer system, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. checking whether a boot configuration information which was created in the preceding boot process exists or not;

C. storing a boot configuration information after execution of the POST operation to a disk, based on the checking result; and

D. loading a graphic user interface (GUI) program.

2. A method according to claim 1, wherein said step C stores the boot configuration information into a disk storage medium.

3. A method according to claim 1, wherein said step C stores the boot configuration information after execution of the POST operation is completed and before an extended memory becomes in use.

4. A method according to claim 1, wherein said step C comprises the steps of:

checking contents of a memory block of a predetermined size;

storing the contents of the memory block into a disc storage medium based on the checking result; and

storing the address of the stored memory block in the disc storage medium.

5. A method according to claim 1, wherein said steps A

to D are performed on a personal computer system.

6. A method for booting a computer system, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. restoring a boot configuration information by using the boot configuration information which has been stored after POST operation in a disk; and

C. loading a graphic user interface (GUI) program.

7. A method according to claim 6, wherein said step B further comprising the steps of:

checking if a designated boot configuration information is different from the restored boot configuration information;

executing an initial driving program based on a modified configuration information; and

updating the boot configuration information after said execution.

8. A method according to claim 6, wherein said step B comprising the steps of:

determining whether to restore said stored boot configuration information;

restoring the contents of memory blocks, addresses of which have been stored in said disk; and

writing zeros into other memory blocks than the restored

memory blocks.

9. A method according to claim 6, wherein said step B restores said stored boot configuration information before an extended memory becomes in use.

10. A method for quickly booting a computer system in which Windows operating system is installed, comprising the steps of:

A. performing a power on self test (POST) of basic input output system (BIOS) when the system is powered on or reset is requested;

B. checking whether a boot configuration information which was created in the preceding boot process exists or not;

C. storing the current boot configuration information in a disk storage medium, if there is no stored boot configuration information;

D. performing a quick POST operation when the computer system is rebooted;

E. restoring the stored boot configuration information from the disk storage medium;

F. updating the boot configuration information before a graphic user interface (GUI) program is loaded, if a designated boot configuration information is different from the restored boot configuration information;

11. A method according to claim 10, wherein said step B calls an interrupt for bootstrap loader to check if the boot configuration information which was created in the preceding

boot process exists.

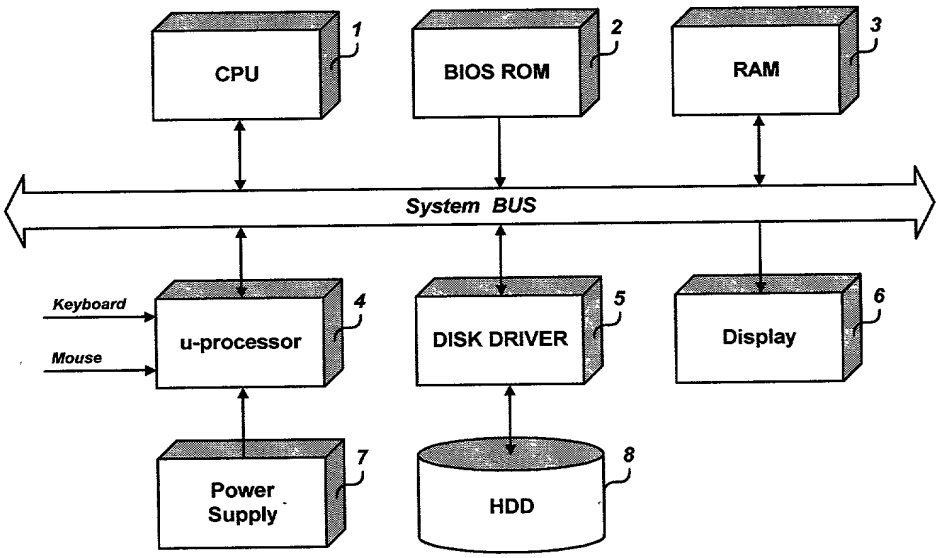
12. A method according to claim 10, wherein said step F determines whether or not the designated boot configuration information is different from the restored boot configuration information based on changes of CONFIG.SYS file and/or AUTOEXEC.BAT file.

ABSTRACT

This invention provides a method for quickly booting a personal computer system by using a boot configuration information on memory and the attached devices that was created and saved in a hard disk at the preceding boot process. The method for a quick boot process according to the present invention comprises the steps of performing a power-on self test (POST) operation when a personal computer system is powered on or a reset button is pressed; performing a normal boot process after the POST operation; saving the contents of memory and the status of the attached devices to a hard disk; checking if a reboot is requested; restoring the saved boot configuration information from the hard disk, after POST is completed during the reboot process; checking whether or not an initial device configuration file and/or an automatic batch file were changed; and executing commands in the two files and saving a newly created boot configuration information to the hard disk for future boot. Accordingly, this invention enables to boot a personal computer system quickly because of omission of execution of the initial device configuration file and the automatic batch file.

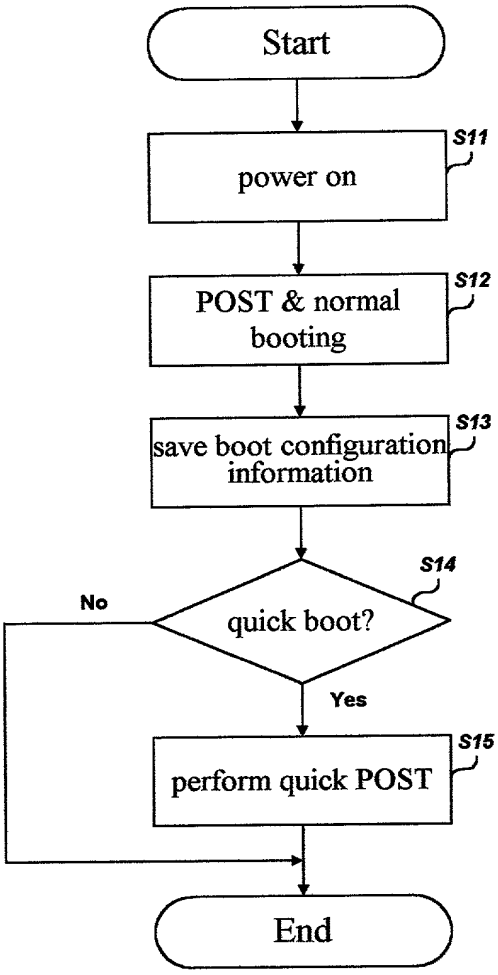
FIGURES

FIG. 1



Conventional Art

FIG. 2



Conventional Art

FIG. 3

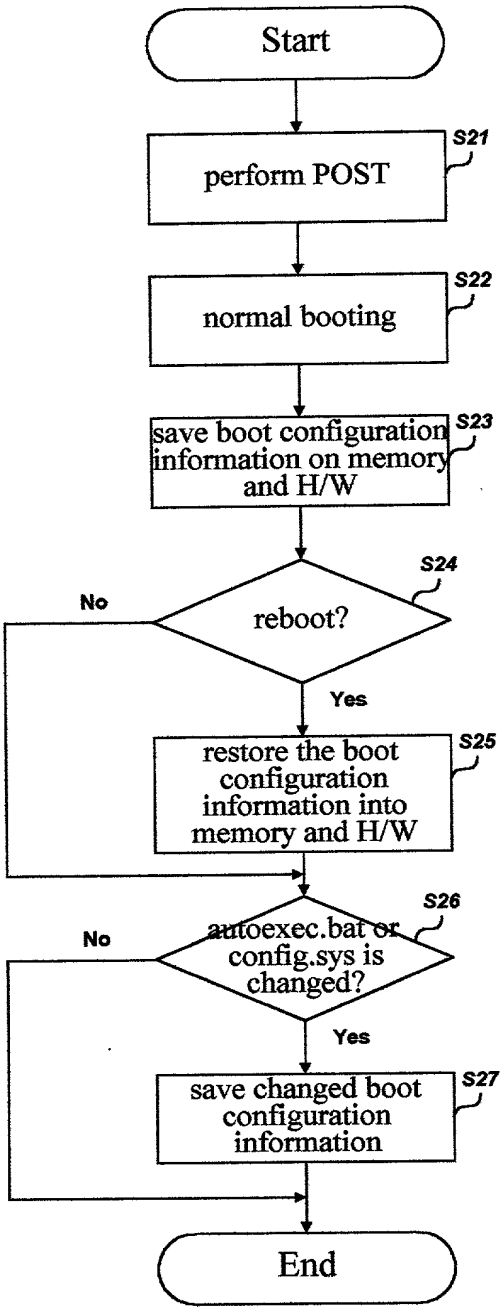

Start
perform POST
S21
normal booting
S22
save boot configuration information on memory and H/W
S23
reboot?
S24
No
Yes
restore the boot configuration information into memory and H/W
S25
autoexec.bat or config.sys is changed?
S26
No
Yes
save changed boot configuration information
S27
End

FIG. 4

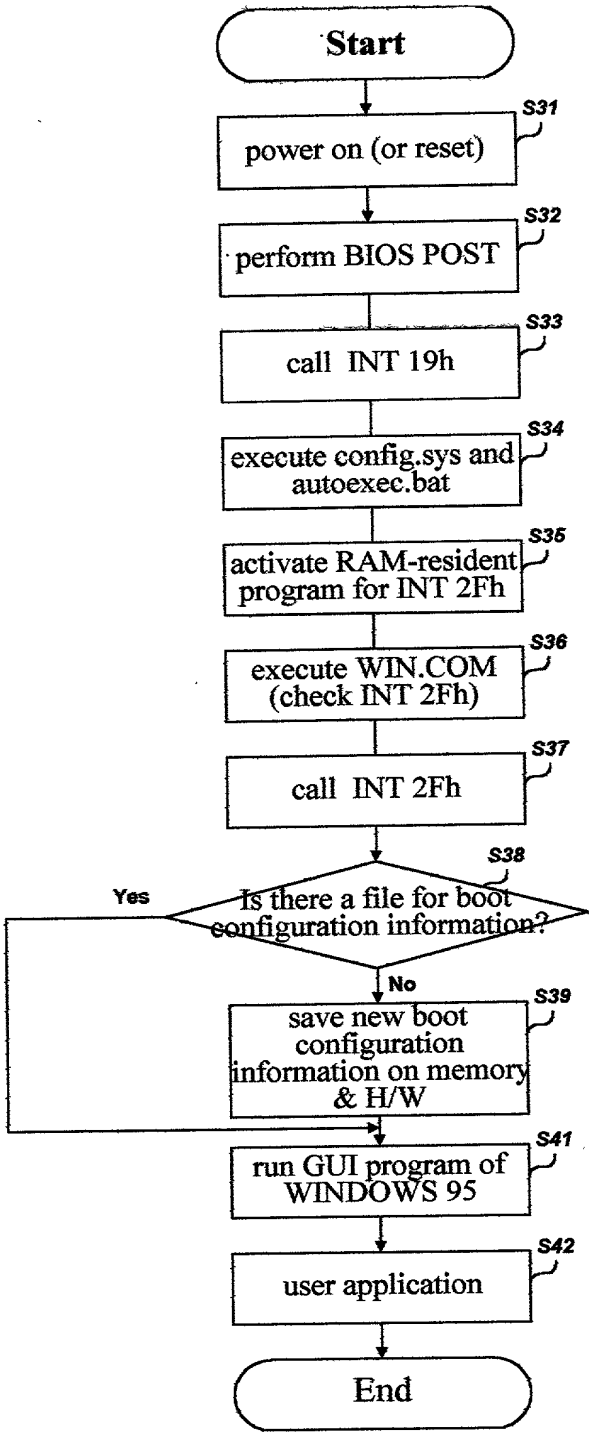

Start
S31
power on (or reset)
S32
perform BIOS POST
S33
call INT 19h
S34
execute config.sys and autoexec.bat
S35
activate RAM-resident program for INT 2Fh
S36
execute WIN.COM (check INT 2Fh)
S37
call INT 2Fh
S38
Is there a file for boot configuration information?
Yes
No
S39
save new boot configuration information on memory & H/W
S41
run GUI program of WINDOWS 95
S42
user application
End

FIG. 5

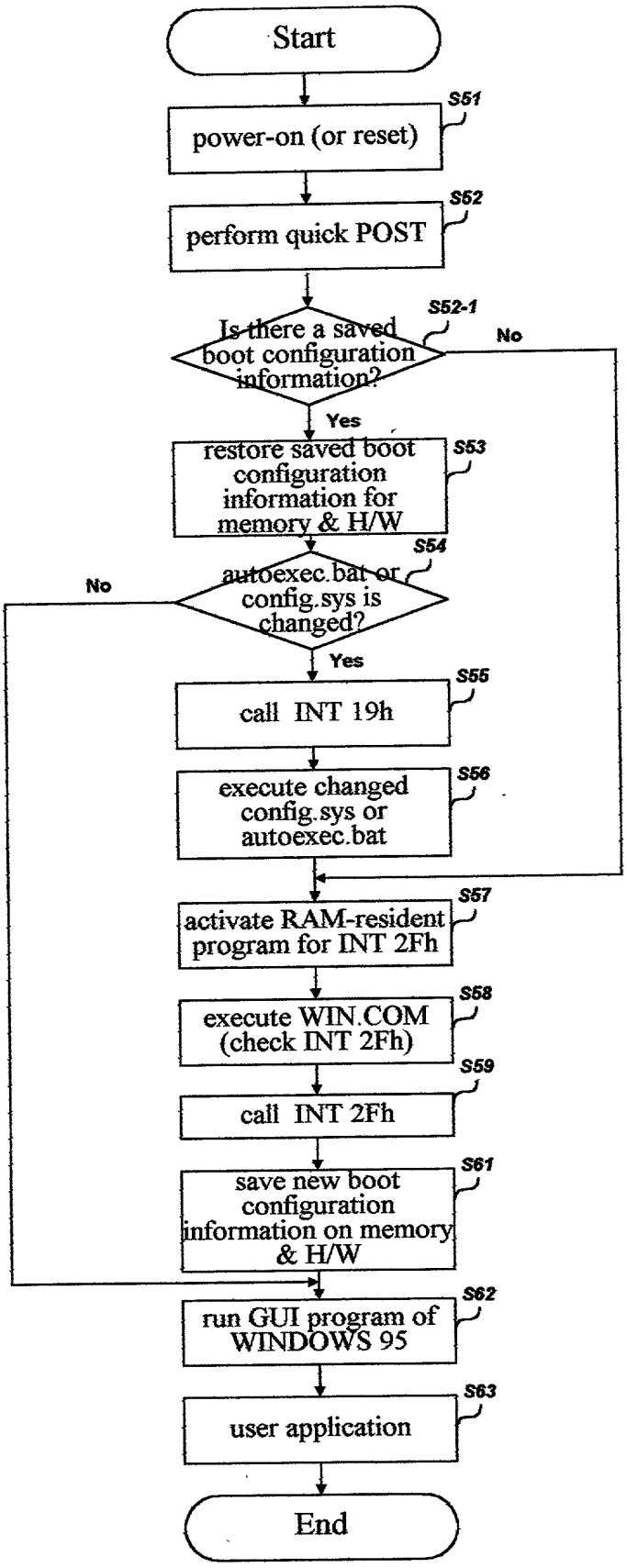

Start
power-on (or reset)
S51
perform quick POST
S52
Is there a saved boot configuration information?
S52-1
No
Yes
restore saved boot configuration information for memory & H/W
S53
autoexec.bat or config.sys is changed?
S54
No
Yes
call INT 19h
S55
execute changed config.sys or autoexec.bat
S56
activate RAM-resident program for INT 2Fh
S57
execute WIN.COM (check INT 2Fh)
S58
call INT 2Fh
S59
save new boot configuration information on memory & H/W
S61
run GUI program of WINDOWS 95
S62
user application
S63
End

FIG. 6

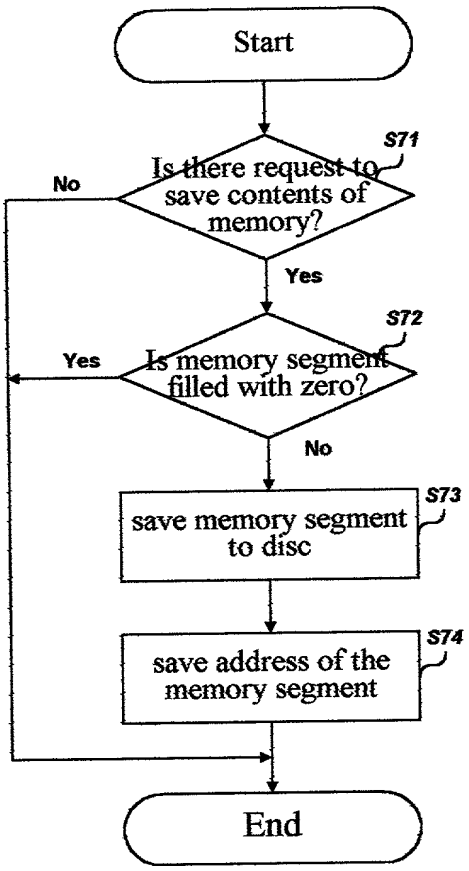
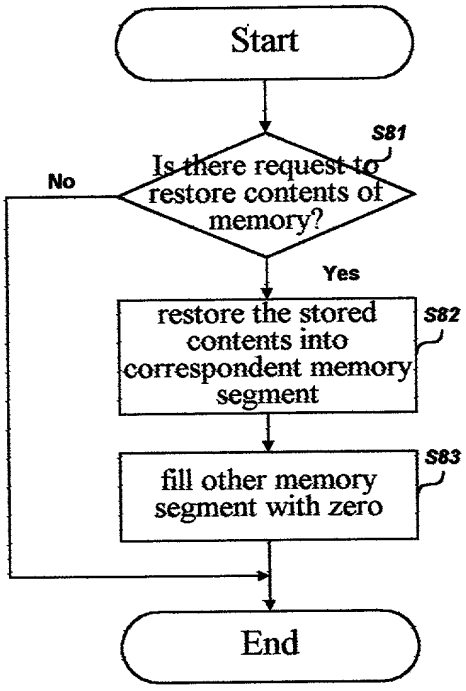

Start
S71
Is there request to save contents of memory?
No
Yes
S72
Is memory segment filled with zero?
Yes
No
S73
save memory segment to disc
S74
save address of the memory segment
End

FIG. 7


Start
S81
Is there request to restore contents of memory?
No
Yes
S82
restore the stored contents into correspondent memory segment
S83
fill other memory segment with zero
End

BIRCH, STEWART, KOLASCH & BIRCH, LLP

PLEASE NOTE: YOU MUST COMPLETE THE FOLLOWING:

COMBINED DECLARATION AND POWER OF ATTORNEY FOR PATENT AND DESIGN APPLICATIONS

ATTORNEY DOCKET NO. 2950-127P

As a below named inventor, I hereby declare that: my residence, post office address and citizenship are as stated next to my name; that I verily believe that I am the original, first and sole inventor (if only one inventor is named below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Insert Title: METHOD FOR QUICKLY BOOTING A COMPUTER SYSTEM

the specification of which is attached hereto. If not attached hereto,

Fill in Appropriate Information - For Use Without Specification Attached:

the specification was filed on _____ as United States Application Number _____; and /or

the specification was filed on _____ as PCT International Application Number _____; and was amended under PCT Article 19 on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56.

I do not know and do not believe the same was ever known or used in the United States of America before my or our invention thereof, or patented or described in any printed publication in any country before my or our invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (six months for designs) prior to this application, and that no application for patent or inventor's certificate on this invention has been filed in any country foreign to the United States of America prior to this application by me or my legal representatives or assigns, except as follows.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 (a)-(d) of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Insert Priority Information: (if appropriate)

Prior Foreign Application(s)

(Number)	(Country)	(Month/Day/Year Filed)	Priority Claimed	
98-17715	Korea	5/11/1998	[X] Yes	[ ] No
			[ ] Yes	[ ] No
			[ ] Yes	[ ] No
			[ ] Yes	[ ] No
			[ ] Yes	[ ] No

I hereby claim the benefit under Title 35, United States Code, §119(e) of any United States provisional application(s) listed below.

Insert Provisional Application(s): (if any)

(Application Number) _____ (Filing Date) _____

(Application Number) _____ (Filing Date) _____

All Foreign Applications, if any, for any Patent or Inventor's Certificate Filed More Than 12 Months (6 Months for Designs) Prior To The Filing Date of This Application:

Insert Requested Information: (if appropriate)

Country	Application No	Date of Filing (Month/Day/Year)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, §1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

Insert Prior U.S. Application(s): (if any)

(Application Number)	(Filing Date)	(Status - patented, pending, abandoned)

I hereby appoint the following attorneys to prosecute this application and/or an international application based on this application and to transact all business in the Patent and Trademark Office connected therewith and in connection with the resulting patent based on instructions received from the entity who first sent the application papers to the attorneys identified below, unless the inventor(s) or assignee provides said attorneys with a written notice to the contrary:

Terrell C. Birch	(Reg. No. 19,382)	Raymond C. Stewart	(Reg. No. 21,066)
Joseph A. Kolasch	(Reg. No. 22,463)	James M. Slattery	(Reg. No. 28,380)
Bernard L. Sweeney	(Reg. No. 24,448)	Michael K. Mutter	(Reg. No. 29,680)
Charles Gorenstein	(Reg. No. 29,271)	Gerald M. Murphy, Jr.	(Reg. No. 28,977)
Leonard R. Svensson	(Reg. No. 30,330)	Terry L. Clark	(Reg. No. 32,644)
Andrew D. Meikle	(Reg. No. 32,868)	Marc S. Weiner	(Reg. No. 32,181)
Joe McKinney Muncy	(Reg. No. 32,334)	Donald J. Daley	(Reg. No. 34,313)
C. Joseph Faraci	(Reg. No. 32,350)		

Send Correspondence to: **BIRCH, STEWART, KOLASCH & BIRCH, LLP**

P.O. Box 747 • Falls Church, Virginia 22040-0747

Telephone: (703) 205-8000 • Facsimile: (703) 205-8050

PLEASE NOTE: YOU MUST COMPLETE THE FOLLOWING:

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of First or Sole Inventor:
Insert Name of Inventor
Insert Date This Document is Signed
Insert Residence
Insert Citizenship
Insert Post Office Address

GIVEN NAME / FAMILY NAME: Seong-Cheol KANG
INVENTOR'S SIGNATURE: [signature]
DATE*: May. 6. 1999
Residence (City, State & Country): Kyunggido, Korea
CITIZENSHIP: Republic of Korea
POST OFFICE ADDRESS (Complete Street Address including City, State & Country): Ra-401, Osan village 37, Gohyun-dong, Osan, Kyunggido, 447-320, Korea

Full Name of Second Inventor, if any:

see above

GIVEN NAME / FAMILY NAME:
INVENTOR'S SIGNATURE:
DATE*:
Residence (City, State & Country):
CITIZENSHIP:
POST OFFICE ADDRESS (Complete Street Address including City, State & Country):

Full Name of Third Inventor, if any

see above

GIVEN NAME / FAMILY NAME:
INVENTOR'S SIGNATURE:
DATE*:
Residence (City, State & Country):
CITIZENSHIP:
POST OFFICE ADDRESS (Complete Street Address including City, State & Country):

Full Name of Fourth Inventor, if any

see above

GIVEN NAME / FAMILY NAME:
INVENTOR'S SIGNATURE:
DATE*:
Residence (City, State & Country):
CITIZENSHIP:
POST OFFICE ADDRESS (Complete Street Address including City, State & Country):

Full Name of Fifth Inventor, if any

see above

GIVEN NAME / FAMILY NAME:
INVENTOR'S SIGNATURE:
DATE*:
Residence (City, State & Country):
CITIZENSHIP:
POST OFFICE ADDRESS (Complete Street Address including City, State & Country):

* DATE OF SIGNATURE